Foundations of Probabilistic Proofs

A course by **Alessandro Chiesa**

Lecture 26

Hardness of Approximation

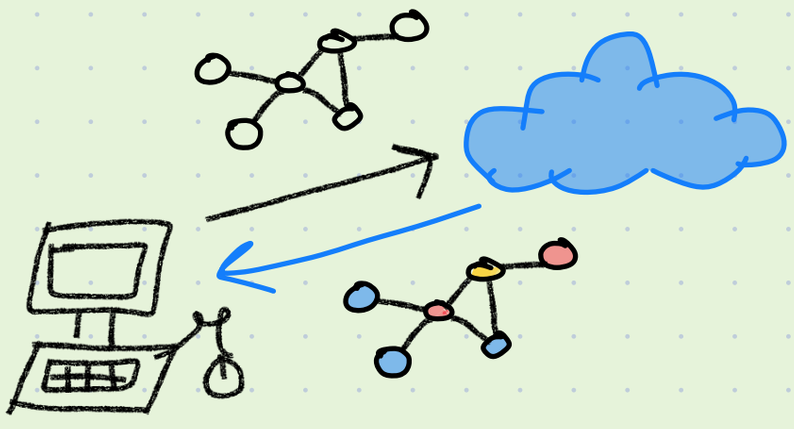
Applications of PCPs

Two main applications of PCPs.

Seen Before

Computational Integrity

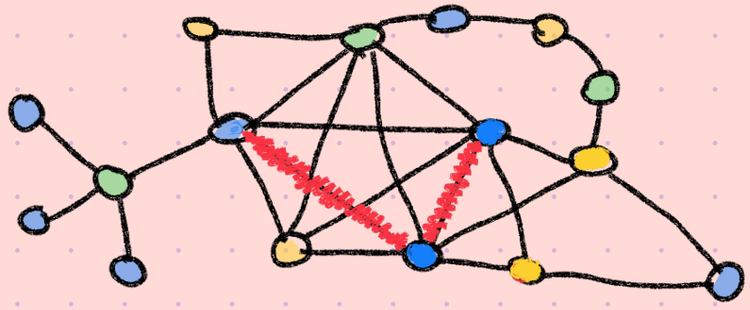
How to verify computations faster than they can be run?



Today

Hardness Of Approximation

Which problems remain hard even if we only seek an approximate solution?



Coping with NP-Hardness

Numerous fundamental optimization problems are NP-hard.

E.g. 3SAT, GRAPHCOLORING, TRAVELINGSALESMAN, KNAPSACK, CLIQUE, VERTEXCOVER, ...

→ If P≠NP then none of these problems can be solved in polynomial time.

Q: How to cope with NP-hardness?

① correct but slow algorithms

Solve the problem exactly via an algorithm whose (super-polynomial) running time is not bad on small instances. Ex: $t(n) = 1.1^n$ ($1.1^{200} < 200$ million $\ll 2^{200}$)

② fast but incorrect algorithms (aka APPROXIMATION ALGORITHMS)

Solve the problem approximately via polynomial-time algorithm.

The quality of the approximation must be guaranteed for every input.

Another direction to cope with NP-hardness is to forego worst-case input guarantees:

ⓘ efficient algorithms that solve "natural" inputs (a major challenge is to model "natural")

ⓘⓘ efficient algorithms that work well in practice (aka heuristics)

Approximation Algorithms

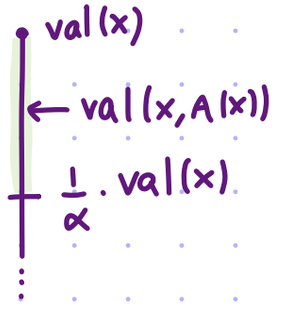
A **maximization problem** is a pair $\mathcal{O} = (sol, val)$ where:

- $sol(x)$ is the (possibly empty) set of valid solutions for the instance $x \in \{0,1\}^*$,
- $val(x,w)$ is the value of the solution $w \in sol(x)$ for x.

We define $val(x) := \max\{val(x,w) : w \in sol(x)\}$.

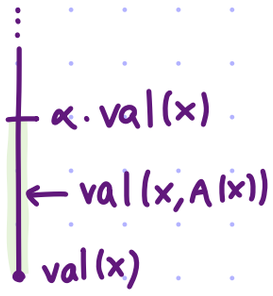
An algorithm A has **approximation ratio $\alpha \geq 1$** for $\mathcal{O} = (sol, val)$ if

$$\forall x \quad val(x, A(x)) \geq \frac{1}{\alpha} \cdot val(x) .$$



The **minimization** case is similar except that:

- $val(x) := \min\{val(x,w) : w \in sol(x)\}$.
- $\alpha \geq 1$ must satisfy $\forall x \quad val(x, A(x)) \leq \alpha \cdot val(x)$.



Example: Max3SAT
- Instances x are 3CNF formulas.
- If x has n variables then $sol(x) = \{0,1\}^n$.
- $val(x,w)$ = fraction of clauses in x satisfied by $w \in \{0,1\}^n$

Basic Goal: design approximation algorithms for NP-hard problems with small α

Approximation Landscape

Researchers have designed approximation algorithms since the 1970s.

Examples:

- Knapsack — $\alpha = 1+\varepsilon$ in time poly$(n, \frac{1}{\varepsilon})$ (FPTAS based on dynamic programming)
- VertexCover — $\alpha = 2$ (vertices in a maximal matching)
- SetCover — $\alpha = O(\log n)$ (pick set covering most points & repeat)
- IndependentSet — α = MaxDegree+1 $= O(n)$ (remove min-degree vertex and its neighbors & repeat)

(SetCover ↑ ↓ IndependentSet: duals!)

- TravelingSalesman — none

Results suggest that problems behave very differently w.r.t. approximation factors.

(Even if the problems are reducible to one another via polynomial-time reductions.)

Q: Why different approximation factors?

NP reductions preserve satisfiability but not necessarily approximability.

They independently transform subgadgets with differing blowups in size.

Hardness of Approximation

Q: Why these specific approximation factors?

We can aim to explain them by showing that one cannot do better:

GOAL: prove hardness of approximation results

We wish to prove statements of the form

"If problem $\mathcal{O}$ is α-approximable in polynomial time then $P = NP$."

(or some other unlikely conclusion (e.g. $3SAT \in DTIME(2^{o(n)})$))

This generalizes the statement "If problem $\mathcal{O}$ is solvable in polynomial time then $P = NP$".

(solvable = 1-approximable)

CHALLENGE: NP-reductions do not have "gaps"

computation of a non-deterministic TM — Cook-Levin Theorem → 3SAT — reduction → problem of interest

(computation of a non-deterministic TM: sensitive to small changes; reduction: local reductions)

→ Strong hardness of approximation requires rejecting computations to be far from accepting.

→ PCPs play a role in hardness of approximation

Constraint Satisfaction Problems

A constraint satisfaction problem (CSPs) generalizes the notion of 3SAT.

def: A (Σ,ℓ,q)-constraint is a pair $C=(S,f)$ where $S\in\binom{[\ell]}{q}$ and $f:\Sigma^S\to\{0,1\}$.
An assignment $a\in\Sigma^\ell$ satisfies $C=(S,f)$ if $f(a(S))=1$.

def: A (Σ,ℓ,q,m)-CSP is a list $\varphi=(C_1,\dots,C_m)$ where each $C_j=(S_j,f_j)$ is a (Σ,ℓ,q)-constraint.
The value of φ on assignment $a\in\Sigma^\ell$ is $val(\varphi,a):=\frac{1}{m}\cdot\sum_{j=1}^m f_j(a(S_j))$.
The value of φ is $val(\varphi):=\max_{a\in\Sigma^\ell} val(\varphi,a)$, and φ is satisfiable if $val(\varphi)=1$.

We associate two problems to CSPs:

- MaxCSP $[\Sigma,\ell,q,m]$ is the search problem that asks to find $a\in\Sigma^\ell$ that maximizes $val(\varphi,a)$
- GapCSP$[\varepsilon_c,\varepsilon_s,\Sigma,\ell,q,m]$ is the decision problem that asks to distinguish if $val(\varphi)\geq 1-\varepsilon_c$ or $val(\varphi)\leq\varepsilon_s$

claim: GapCSP$[\varepsilon_c,\varepsilon_s,\Sigma,\ell,q,m]$ is NP-hard → approximating MaxCSP$[\Sigma,\ell,q,m]$ with $\alpha<\frac{1-\varepsilon_c}{\varepsilon_s}$ is NP-hard

proof: If $val(\varphi)\geq 1-\varepsilon_c$ then the approximation is at least $\frac{1}{\alpha}\cdot val(\varphi)\geq\frac{1}{\alpha}\cdot(1-\varepsilon_c)>\frac{\varepsilon_s}{1-\varepsilon_c}\cdot(1-\varepsilon_c)=\varepsilon_s$.
If $val(\varphi)\leq\varepsilon_s$ then the approximation is at most $val(\varphi)\leq\varepsilon_s$. ■

To show hardness of approximation it suffices to show hardness of gap problems.

CSP vs PCP

There is an equivalence between CSPs and (non-adaptive) PCPs.

claim: L reduces to $\mathrm{GapCSP}[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, m] \to L \in \mathrm{PCP}[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, r=\log m]$

proof:

$V_{PCP}^{a \in \Sigma^\ell}(x)$: 1. Reduce x to the CSP $\varphi = (C_1, \ldots, C_m)$.

2. Sample $j \in [m]$.

3. Check that $f_j(a(S_j)) = 1$.

The PCP verifier makes q queries and uses $r = \log m$ random bits.

Completeness and soundness follow from the fact that $\forall a \in \Sigma^\ell \ \Pr[V_{PCP}^a(x)=1] = \mathrm{val}(\varphi, a)$. ■

claim: $L \in \mathrm{PCP}[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, r] \to L$ reduces to $\mathrm{GapCSP}[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, m=2^r]$ in time $\mathrm{poly}(2^r, n)$

proof: Let $V = (S, Q, D)$ be the (non-adaptive) PCP verifier for L.

Map the instance x to the CSP instance $\varphi = (C_\rho)_{\rho \in \{0,1\}^r}$ where

$$C_\rho = (S_\rho, f_\rho) \text{ with } S_\rho = Q(x, \rho) \text{ and } f_\rho(a(S_\rho)) = D(S(x,\rho), a(S_\rho)).$$ ■

DICTIONARY:

PCP verifier	CSP instance	proof length	number of variables
PCP string	assignment	query complexity	arity of constraints
completeness/soundness	YES/NO thresholds	randomness complexity	log of number of constraints

Inapproximability of Max3SAT [1/4]

We have learned that

$$NP \subseteq PCP[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, r = O(\log n)]$$

$$\rightarrow \text{GapCSP}[\varepsilon_c, \varepsilon_s, \Sigma, \ell, q, m = \text{poly}(n)] \text{ is NP-hard}$$

$$\rightarrow \text{approximating MaxCSP}[\Sigma, \ell, q, m] \text{ with } \alpha < \frac{1-\varepsilon_c}{\varepsilon_s} \text{ is NP-hard}$$

Hence the PCP Theorem tell us that

$$\exists\, \varepsilon_s \in (0,1)\ \exists\, q \in \mathbb{N} \text{ s.t. GapCSP}[\varepsilon_c = 0, \varepsilon_s, \Sigma = \{0,1\}, \ell = \text{poly}(n), q, m = \text{poly}(n)] \text{ is NP-hard.} \quad \Delta$$

What can we say about the inapproximability of Max3SAT? (MaxCSP with $q=3$ and each f_j is an OR-clause of ≤ 3 literals from S_j)

theorem: $\exists\, \varepsilon_s \in (0,1)$ s.t. deciding if a 3SAT φ has $\text{val}(\varphi)=1$ or $\text{val}(\varphi) \leq \varepsilon_s$ is NP-hard

The Cook-Levin theorem tells us that for a 3SAT φ it's NP-hard to distinguish $\text{val}(\varphi)=1$ and $\text{val}(\varphi) \leq 1 - \frac{1}{m}$.

The above theorem significantly strengthens this:

the YES/NO gap increases from $\frac{1}{m}$ (inverse polynomial) to $1-\varepsilon_s$ (constant).

The theorem is a direct consequence of Δ and the lemma in the next slide, which reduces GapCSP to Gap3SAT by expressing each constraint as a 3CNF of size $2^q \cdot q$. (constant if $q = O(1)$)

Inapproximability of Max3SAT [2/4]

lemma: $\mathrm{GapCSP}[\varepsilon_c, \varepsilon_s, \Sigma=\{0,1\}, \ell, q, m]$ reduces to $\mathrm{Gap3SAT}[\varepsilon_c, \varepsilon_s' = 1-\frac{1-\varepsilon_s}{2^q \cdot q}, \Sigma=\{0,1\}, \ell' = \ell + m2^q q, q'=3, m' = m2^q q]$

proof: Let $\varphi = (C_1, \ldots, C_m)$ be a $(\{0,1\}, \ell, q, m)$-CSP.

AND-OR encoding of a boolean function

For each $j \in [m]$, express the constraint $C_j = (S_j, f_j : \{0,1\}^{S_j} \to \{0,1\})$ as a boolean formula $\bigwedge_{b \in \{0,1\}^{S_j}} \varphi_{j,b}$ where each $\varphi_{j,b}$ is an OR-clause over $(x_i)_{i \in S_j}$ that encodes the b-th row in f_j's evaluation table as follows:

- if $f_j(b) = 1$ then $\varphi_{j,b}((x_i)_{i \in S_j}) := 1$ (always TRUE),
- if $f_j(b) = 0$ then $\varphi_{j,b}((x_i)_{i \in S_j}) := \bigvee_{i \in S_j} (x_i \oplus b_i)$ (i.e. $\varphi_{j,b}$ outputs 1 precisely on inputs $\neq b$).

Note that $\forall\, c \in \{0,1\}^{S_j}$ $f_j(c) = 1$ iff c satisfies $\bigwedge_{b \in \{0,1\}^{S_j}} \varphi_{j,b}$.

Next express each $\varphi_{j,b}$ as a 3CNF $\bigwedge_{t \in [q]} \varphi_{j,b,t}$ using q clauses and q auxiliary variables $(x_{j,b,t})_{t \in [q]}$.

Define the 3CNF formula

$$\varphi' := \bigwedge_{j \in [m]} \bigwedge_{b \in \{0,1\}^{S_j}} \bigwedge_{t \in [q]} \varphi_{j,b,t},$$

which is over the $\ell' := \ell + m \cdot 2^q \cdot q$ variables $(x_i)_{i=1}^{\ell}$ and $(x_{j,b,t})_{j \in [m],\, b \in \{0,1\}^{S_j},\, t \in [q]}$ and has $m' := m \cdot 2^q \cdot q$ clauses.

Fix $a \in \{0,1\}^{\ell}$ and $j \in [m]$.

- If $f_j((a_i)_{i \in S_j}) = 1$ then $\exists\, (a_{j,b,t})_{b \in \{0,1\}^{S_j},\, t \in [q]}$ s.t. $\bigwedge_{b \in \{0,1\}^{S_j}} \bigwedge_{t \in [q]} \varphi_{j,b,t}(a, (a_{j,b,t})_{t \in [q]}) = 1$.
- If $f_j((a_i)_{i \in S_j}) = 0$ then $\forall\, (a_{j,b,t})_{b \in \{0,1\}^{S_j},\, t \in [q]}$ $\exists\, b^*, t^*$ s.t. $\varphi_{j,b^*,t^*}(a, (a_{j,b,t})_{t \in [q]}) = 0$.
  (and possibly $\forall\, b\ \forall\, t \cdot \varphi_{j,b,t}(a, (a_{j,b,t})_{t \in [q]}) = 0$)

We conclude that $\mathrm{val}(\varphi') \geq \mathrm{val}(\varphi)$ and $1 - \mathrm{val}(\varphi') \geq \frac{1-\mathrm{val}(\varphi)}{2^q \cdot q}$ (i.e. $\mathrm{val}(\varphi') \leq 1 - \frac{1-\mathrm{val}(\varphi)}{2^q \cdot q}$). ■

Inapproximability of Max3SAT [3/4]

lemma: ∃ expected polynomial-time algorithm with approximation ratio $\alpha = \frac{8}{7}$ for MaxE3SAT [David S. Johnson 1974]

formula is an E3CNF
(each clause has exactly 3 literals)

proof:

$A(\varphi)$: 1. Sample random $a \in \{0,1\}^n$.

2. If $val(\varphi,a) < 7/8$ go to 1. Else output a.

The algorithm A outputs a s.t. $val(\varphi,a) \geq \frac{7}{8} = \frac{1}{8/7} \cdot 1 \geq \frac{1}{8/7} \cdot val(\varphi)$.

We are left to analyze its expected running time.

For $j \in [m]$, $Z_j :=$ "indicator that a random $a \in \{0,1\}^n$ satisfies j-th clause of φ".

Note that $\mathbb{E}[Z_j] = \Pr[Z_j = 1] = 1 - \Pr[Z_j = 0] = 1 - \frac{1}{2}\cdot\frac{1}{2}\cdot\frac{1}{2} = 7/8$. (Each clause has exactly 3 literals.)

Hence $\mathbb{E}[\sum_{j=1}^{m} Z_j] = \sum_{j=1}^{m} \mathbb{E}[Z_j] = 7/8 \cdot m$.

We deduce that $\exists\, a \in \{0,1\}^n$ that satisfies $7/8 \cdot m$ clauses.

In fact, $p := \Pr[\sum_{j=1}^{m} Z_j \geq 7/8 \cdot m] \geq \frac{1}{8m}$ because, letting $p_k := \Pr[\sum_{j=1}^{m} Z_j = k]$,

$$\frac{7}{8}\cdot m = \mathbb{E}\left[\sum_{j=1}^{m} Z_j\right] = \sum_{0 \leq k \leq \lfloor \frac{7}{8}m - 1 \rfloor} k \cdot p_k + \sum_{\frac{7}{8}m \leq k \leq m} k \cdot p_k \leq \left(\frac{7}{8}m - \frac{1}{8}\right) \sum_{0 \leq k < \frac{7}{8}m} p_k + m \cdot \sum_{\frac{7}{8}m \leq k \leq m} p_k \leq \left(\frac{7}{8}m - \frac{1}{8}\right)\cdot 1 + m \cdot p \quad \longrightarrow \quad p \geq \frac{1}{8m}.$$

So the expected number of samples is $8 \cdot m$, and thus A runs in expected time $O(m^2)$. ■

Inapproximability of Max3SAT [4/4]

The expected-time algorithm can be derandomized via the method of CONDITIONAL EXPECTATIONS, yielding an $O(n \cdot m)$-time algorithm (even $O(n+m)$ with some care).

For $i=1,\dots,n$: set $x_i := b_i$ to maximize $\mathop{\mathbb{E}}_{a}[\mathrm{val}(\varphi,a) \mid a_1=b_1,\dots,a_{i-1}=b_{i-1}]$, which is computable in time $O(m)$.

We showed that $\exists\, \varepsilon_s \in (0,1)$ s.t. distinguishing if a E3CNF φ has $\mathrm{val}(\varphi)=1$ or $\mathrm{val}(\varphi) \leq \varepsilon_s$ is NP-hard.

How small can ε_s be?

We expect $\varepsilon_s \geq \frac{7}{8}$ due to the $\frac{8}{7}$-approximation algorithm for MaxE3SAT.

Some Optimal Inapproximability Results

Johan Håstad
KTH

This lower bound can be matched:

<u>theorem:</u> $\forall \gamma > 0$ it is NP-hard to distinguish if a E3CNF φ has $\mathrm{val}(\varphi)=1$ or $\mathrm{val}(\varphi) \leq \frac{7}{8}+\gamma$

The theorem implies that the approximation ratio $\alpha = \frac{8}{7}$ for MaxE3SAT cannot be increased (assuming $P \neq NP$).

The proof shows that $\forall \gamma > 0$ $NP \subseteq PCP\left[\varepsilon_c = 0,\ \varepsilon_s \leq \frac{7}{8}+\gamma,\ \Sigma=\{0,1\},\ \ell = \mathrm{poly}(n),\ q=3,\ r=O(\log n)\right]$.

where the (non-adaptive) PCP verifier decides via an E3CNF clause.

Tools include: PCP Theorem, parallel repetition, long code, Fourier analysis of boolean functions.

There is a (deterministic polynomial-time) $\frac{8}{7}$-approximation algorithm for Max3SAT, [Howard Karloff, Uri Zwick 1997]

so the gap in the theorem cannot be improved for Max3SAT (≤ 3 literals) compared to MaxE3SAT ($=3$ literals).

Inapproximability of Other MaxCSP

Other CSPs have other approximation ratios and (sometimes matching) inapproximability results.

EXAMPLE: MaxB3CSP := MaxCSP[$\Sigma=\{0,1\}, \ell, q=3, m$].

(Relaxation of Max3SAT where each predicate $f_j:\{0,1\}^{s_j}\to\{0,1\}$ can be arbitrary, rather than being required to be a 3CNF clause.)

There is a (deterministic polynomial-time) $\frac{8}{5}$-approximation algorithm for MaxB3CSP. [Zwick 1998]

This approximation ratio cannot be decreased (unless P≠NP):

theorem: $\forall \gamma>0 \quad NP \subseteq PCP[\varepsilon_c=0, \varepsilon_s\leq\frac{5}{8}+\gamma, \Sigma=\{0,1\}, \ell=poly(n), q=3, r=O(\log n)]$

↑ the non-adaptive PCP verifier can be arbitrary (its decision predicate need not rule according to a 3CNF clause)

On the NP-Hardness of MAX-NOT-2

Johan Håstad
KTH

Bibliography

Hardness of approximation

- [FGLSS 1996]: Interactive proofs and the hardness of approximating cliques, by Uriel Feige, Shafi Goldwasser, Laszlo Lovasz, Shumel Safra, Mario Szegedy.
- [ACY 2022]: Hardness of approximation for stochastic problems via interactive oracle proofs, by Gal Arnon, Alessandro Chiesa, Eylon Yogev.

Inapproximability of 3SAT and other 3CSPs

- [Håstad 1997]: Some optimal inapproximability results, by Johan Håstad.
- [GS 2005]: Hardness of Max 3SAT with no mixed clauses, by Venkatesan Guruswami, Subhash Khot.
- [Håstad 2012]: On the NP-Hardness of Max-Not-2, by Johan Håstad.